Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

CRT-2006-12

A SIMULATION-BASED DYNAMIC TRAFFIC ASSIGNMENT MODEL : DYNAMEQ

by

Michael Florian
Michael Mahut
Nicolas Tremblay

June 2006

Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal QC H3C 3J7 Canada
Téléphone: (514) 343-7575 / Télécopieur: (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca

HEC · ÉCOLE POLYTECHNIQUE MONTRÉAL · Université de Montréal

106-07233/

PUBLICATION
CRT-2006-12

A SIMULATION-BASED DYNAMIC TRAFFIC ASSIGNMENT MODEL : DYNAMEQ

by

Michael Florian[1,2]
Michael Mahut[1,2]
Nicolas Tremblay[2]

The research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

1. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 (mike@crt.umontreal.ca)
2. Les Conseillers INRO Inc., 5160, boulevard Décarie, Montréal, Canada H3X 2H9 (michaelm,nicolast@inro.ca)

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2006

ABSTRACT

This paper describes the model, solution algorithm and some selected applications of a simulation-based dynamic traffic assignment model based on equilibrium concepts. The model is formulated as a variational inequality and a solution algorithm is developed for a discretized version of the model. It is most useful, in its current form, for off-line applications, such as the testing of control strategies and scenario analyses in planning studies that must consider temporal flow variations and are not well served by the application of static models. The determination of time-dependent path input flows is modeled as a master problem inspired from a simplicial decomposition approach. The determination of path travel times for a given set of path flows is the network-loading sub-problem, which is solved using the simplified traffic simulation approach of Mahut. The reported applications of this model include references to studies carried out in several cities around the world and some detailed model calibration results for Montreal. The results show that this model is applicable to medium-size networks with very reasonable computing times

Key words: Dynamic traffic assignment, Method of successive averages, Traffic simulation, Queuing models

RÉSUMÉ

Cet article présente la formulation, l'algorithme de solution et certaines applications d'un modèle d'affectation dynamique du trafic basé sur des notions d'équilibre et qui utilise la simulation pour représenter l'évolution temporelle du trafic. Le modèle est formulé sous la forme d'une inégalité variationnelle et l'algorithme de solution est développé pour une version discrète du modèle. Le modèle est utile, dans sa forme actuelle, pour des applications de planification qui doivent considérer des variations temporelles du flot qui ne sont pas offertes par des modèles statiques. La détermination des flots temporels d'entrée des chemins est considérée comme un problème maître inspiré d'une approche de décomposition simpliciale. Le chargement du réseau est accompli par une simulation simplifiée du trafic qui est basée sur l'approche de Mahut. Les applications du modèle incluent des références aux études accomplies dans plusieurs villes du monde et des résultats de calibration détaillés pour la ville de Montréal. Les résultats démontrent que le modèle est applicable à des réseaux de taille moyenne demandant un temps de calcul raisonnable.

Mots clés: Affectation dynamique du trafic, méthode des moyennes successives, simulation du trafic, modèles de files d'attente.

1. INTRODUCTION

This paper provides a summary description and some selected applications of an equilibrium based dynamic traffic assignment model. In order to provide a common terminology to the various models, it is convenient to refer to the main components of any dynamic traffic model: the route-choice mechanism, the determination of the path input flows and the network-loading mechanism. The latter is the method used to represent the evolution of the traffic flow over the links of the network once the route choice and the path input flows have been determined.

Some of the most popular dynamic traffic models today are those based on the representation of the behavior of each driver regarding car following, gap acceptance and lane choice. These are micro-simulation models. The successful use of micro-simulations is commonly limited to relatively small size networks. Their application has been hindered for medium-to-large networks by the relatively high computation time and the effort required for a proper model calibration. Nevertheless, these models are popular and their use is enhanced by traffic animation graphics that capture the attention of non-technical staff. Some of these micro-simulation models are

CORSIM (http://www.fhwa-tsis.com/corsim_page.htm),
INTEGRATION (Van Aerde, 1999),
AIMSUN2 (Barceló et al, 1994) (http://www.tss-bcn.com),
VISSIM (http://www.ptv.de),
PARAMICS (http://www.quadstone.com)
and DRACULA (http://www.its.leeds.ac.uk/software/dracula/).
MITSIM (Yang, 1997) (http://web.mit.edu/its/products.html) is an academic research model that has been used in several studies in Boston, Stockholm and elsewhere.

The aim of handling larger networks with reasonable computational times has led to the development of so-called "mesoscopic" approaches to traffic simulation. The aim is to obtain a traffic representation that still captures the basic temporal congestion phenomena, but models the traffic dynamics with less fidelity. Some of the earliest examples of such an approach are CONTRAM (Leonard et. al., 1989) (www.contram.com) and SATURN (Dow and Van Vliet, 1979) which are commercially available packages that are still used in England and elsewhere.

Recently, the development of mesoscopic simulation models for off-line dynamic traffic assignment has become an area of significant research activity, as witnessed by the United States FHWA Dynamic Traffic Assignment Project (http://www.dynamictrafficassignment.org). The development of DYNASMART (Mahmassani et al., 2001) and DYNAMIT (Ben-Akiva et al., 1998) (http://web.mit.edu/) are two significant developments. Another approach to the network loading algorithm is that based on cellular automata theory (Nagel and Schreckenberg, 1992), which has been implemented in the TRANSIMS software (http://transims.tsasa.lanl.gov), developed by the Los Alamos National Laboratories in the USA. The advance of vehicles is carried out by using local rules for each vehicle that determine the next cell to be occupied and the speed of the vehicle. Ziliaskopoulos and Lee (1997) developed a dynamic traffic assignment

model based on the cell transmission model which is a particular solution method for the classical macroscopic traffic flow model.

Other dynamic traffic assignment models that have their roots in macroscopic traffic flow theory (Lighthill and Whitham, 1955) (Richards, 1956) such as METACOR (Diakakis and Papageorgiou, 1996) and METANET (Messmer et al., 2000a), These models are based on the work of Papageorgiou (1990) and Messmer et al., 2000b respectively. The route choice in METANET is carried out iteratively by splitting proportions at nodes of the network, with the restriction that only two arcs can originate at a given node. The network loading method is based on a second order (p.d.e.) traffic flow model.

Another line of research is that of analytical dynamic traffic assignment models, which have their roots in the mathematical programming and optimal control approaches. Their aim is to introduce temporal dimensions to extensions of static network equilibrium models which are based on link travel time functions. There is a very large body of literature that contains academic contributions made by using this approach. A good recent reference is a special issue of *Networks and Spatial Economics* (Vol.1, Issue 3/4, 2001). The paper by Friesz et al (1993) provides a formulation the equilibrium dynamic traffic model which serves as the basis for the algorithm developed here.

The network loading method presented in this paper is based on a traffic simulation model that was designed to produce reasonably accurate results with a **minimum** number of parameters and a **minimum** of computational effort (Mahut, 2000, Astarita et al., 2001)). The underlying structure of the network model and the car mover have more in common with microscopic than with mesoscopic approaches, as the model is designed to capture the effects of car following, lane changing and gap acceptance.

The paper is structured as follows. The next section is dedicated to the exposition of the network and demand representation used in the model; then the formal statement of the model and its discretized version are given. The next sections are dedicated to the statement of the algorithm and to the description of the network loading method. Applications of the model are then given and some conclusions end the paper.

2. NETWORK REPRESENTATION, TRAFFIC CONTROL AND DEMAND

The network definition required for this DTA model requires somewhat more information than that required for static network equilibrium models, yet somewhat less than is generally required for micro-simulation traffic models.

Since the underlying traffic model moves individual vehicles on discrete lanes, each link must be defined by a number of lanes. Each lane furthermore is defined by an access code that determines which classes of vehicles may use the lane (e.g., taxi, bus, HOV, etc…). A length and speed limit furthermore define each link. At each node (intersection) of the network, a turn is defined for each permitted movement from an incoming link to an outgoing link. Each turn is defined by an access code and a saturation flow rate per lane. Unlike micro-simulation models, the network definition does not require geometrical information such as lane width, turning angles, and the dimensions of intersections.

The model also permits the specification of detailed traffic control information such as (pre-timed) signal timing and ramp metering plans. Traffic control specifications furthermore

require the number of lanes associated with each turning movement, and the lanes (on both the upstream and downstream links) that may be used for executing a turn. These data may vary with the signal phase rather than being fixed for each turn.

Vehicle attributes (or parameters) can be broken down into two distinct categories: physical attributes and routing attributes. The physical attributes are the effective length (based on vehicle spacing at jam density), and the driver/vehicle response time. Together, these parameters yield the jam density and negative wave speed associated with each vehicle class.

Routing attributes include the vehicle class identifier, which determines which lanes and turns of the network may be used by the class, and identifies any class-based routing strategy that may be defined. For instance, the class car uses different routing rules than the class bus, which travels along fixed itineraries and has mandatory stops. A demand matrix by class contains the flow in vehicles per hour for each origin-destination pair. The matrices are "time-sliced" in the sense that flow rates may be specified for given time intervals. The demand is considered to be fixed in this version of the model.

3. DYNAMIC TRAFFIC ASSIGNMENT – THE MODEL

Two different approaches are commonly used to emulate the path choice behavior of drivers: dynamic assignment *en route* and dynamic *equilibrium* assignment. In this work, the approach taken is to seek an approximate solution to the dynamic equilibrium conditions. Extensions of the model for *en route* assignment are indicated at the conclusion of the paper.

In the equilibrium assignment problem, only pre-trip path choices are considered. The path choices are modeled as decision variables governed by a user optimal principle where each driver seeks to minimize the used path travel time. All drivers have perfect access to information, which consists of the travel times on all paths (used and unused). The solution algorithm takes the form of an iterative procedure designed to converge to these conditions.

The mathematical statement of the dynamic equilibrium problem is in the space of path flows $h_k(t)$, for all paths k belonging to the set K_i for an origin-destination $i \in I$, at time t. The time-varying demands are denoted $g_i(t)$. The path flow rates in the feasible region Ω satisfy the conservation of flow and non-negativity constraints for $t \in T_d$, where $(0, T_d)$ is the period during which the temporal demand is defined. That is

$$\Omega = h(t) \cdot \sum_{k \in K_i} h_k(t) = g_i(t), i \in I; h_k(t) \geq 0 \quad \textit{for almost all } t \in T_d. \tag{1}$$

The definition of user optimal dynamic equilibrium is given by the temporal version of the static (Wardrop) user optimal equilibrium conditions, which are:

$$\begin{aligned} &s_k(t) = u_i(t) \quad \textit{if} \quad h_k(t) > 0 \\ &s_k(t) \geq u_i(t) \quad \textit{otherwise} \end{aligned} \qquad \text{for all } k \in K_i, \text{ for almost all } t \in T_d \tag{2}$$

where: $h_k \in \Omega, u_i(t) = \min_{k \in K_i} \{s_k(t)\}$ for almost all $t \in T_d$ and $s_k(t)$ is the path travel time of path k.

Friesz et al (1993) showed that these conditions are equivalent to a variational inequality problem, which is to find $h^* \in \Omega$ such that

$$\left(S(h^*),\ h - h^*\right) \geq 0, \forall h \in \Omega \tag{3}$$

The solution approach adopted for solving the dynamic network equilibrium model (1)-(3) is based on a temporal discretization into discrete periods $\tau = 1, 2, ..., \left|\frac{T_d}{\Delta t}\right|$, where Δt is the chosen duration of a time interval. This results in the discretized model

$$\begin{array}{l} s_k^\tau = u_i^\tau \text{ if } h_k^\tau > 0 \\ s_k^\tau \geq u_i^\tau \text{ if } h_k^\tau = 0 \end{array} \quad \text{for all } k \in k_i,\ i \in I, \tau = 1, 2, ..., \left|\frac{T_d}{\Delta t}\right| \tag{4}$$

where the feasible set of time dependent flows h_k^τ belong to

$$\Omega^\tau = h_k^\tau : \sum_{k \in K_i} h_k^\tau = g_i^\tau, i \in I, \text{all } \tau; \tag{5}$$

$$h_k^\tau \geq 0, k_i, i \in I, \text{all } \tau,$$

which can be shown to be equivalent to solving the discretized variational inequality.

$$\sum_\tau \sum_{k \in K} s_k^\tau(h^\tau)\,(h_k^\tau - h_k^\tau) \geq 0 \tag{6}$$

where $K = \bigcup_{i \in I} k_i$ where h^τ is the vector of path flows (h_k^τ) for all k and τ.

The demonstration of existence and uniqueness of a solution to this model depends on the properties of the mapping ***s(h[g])***, that is the dependence of link and path travel times on the path input flows and the dependence of the path input flows on the demands. Since the properties of this mapping are not easily verified due to the fact that it is the output of a simulation model and not an analytical transformation, no claims are made about the existence or the uniqueness of a solution. The equilibrium principle is used as a **guide** in computing an approximate solution of the time discrete variational inequality.

4. THE DTA ALGORITHM

The solution algorithm used here consists of two main components other than the computation of the temporal shortest paths: a method to determine a new set of time-dependent path input flows, given the experienced path travel times at the previous iteration, and a method to determine the actual link flows and travel times that result from a given set of path inflow rates.. The algorithm furthermore requires a set of initial path flows. The general structure of the algorithm is shown schematically in Figure 1.

The path input flows $h_k^\tau, k \in K$ are determined by a variant of the method of successive averages (MSA), which is applied to each O-D pair I and time interval τ. An initial feasible solution is computed by assigning the demand for each time period to a set of successive shortest paths. Starting at the second iteration, and up to a pre-specified maximum number of iterations, N, the time-dependent link travel times after each loading are used to determine a new set of dynamic shortest paths that are added to the current set of paths iteration $n, n \le N$, the volume assigned as input flow to each path in the set is g_i^τ / n, $i \in I$, *all* τ. After that, for iteration $m, m > N$, only the shortest among *used* paths is identified and the path input flow rates are redistributed over the known paths.

If the flow of a particular path decreases below a small predetermined value then the path is dropped and its remaining flow is distributed to the other used paths. This heuristic approach is akin to the restricted simplicial decomposition algorithm of Lawgphonpanich and Hearn (1987) for the solution of the static network equilibrium model with fixed demand. The algorithm is summarized below:

EQUILIBRIUM DTA ALGORITHM

- *Step 0 Initialization (l=1):*
- Compute temporal shortest paths based on free-flow travel times;
- load the demands to obtain an initial solution; $l = l + 1$
- *Step 1 Reallocation of input flows to paths:*
  - *Step 1.1* If $l \le N$

    Compute a new dynamic shortest path;
    assign to each path k the input flow g_i^τ / l

  Step 1.2 If $l > N$

    Identify the shortest among used paths;
    redistribute the flows as follows:

$$h_k^l(\tau) = \begin{cases} h_k^{l-1}(\tau)\left(\frac{l-1}{l}\right) + \frac{g_i(\tau)}{l}, & \text{if } s_k^l(\tau) = u_k^l(\tau) \\ h_k^{l-1}(\tau)\left(\frac{l-1}{l}\right) & \text{otherwise} \end{cases} \qquad (7)$$

- *Step 2 Stopping rule:*

  If $l \le L$ *or* $NGap \le \varepsilon \Rightarrow STOP$;

  otherwise return to Step 1

While no formal convergence proof can be given for this algorithm, since the network loading map does not have an analytical form, a measure of gap, inspired from that used in static network equilibrium models may be used for qualifying a given solution. It is the difference between the total travel time experienced and the total travel time that would

have been experienced if all vehicles had the travel time (over each interval τ) equal to that of the current shortest path. Other step size rules can be easily implemented as well.

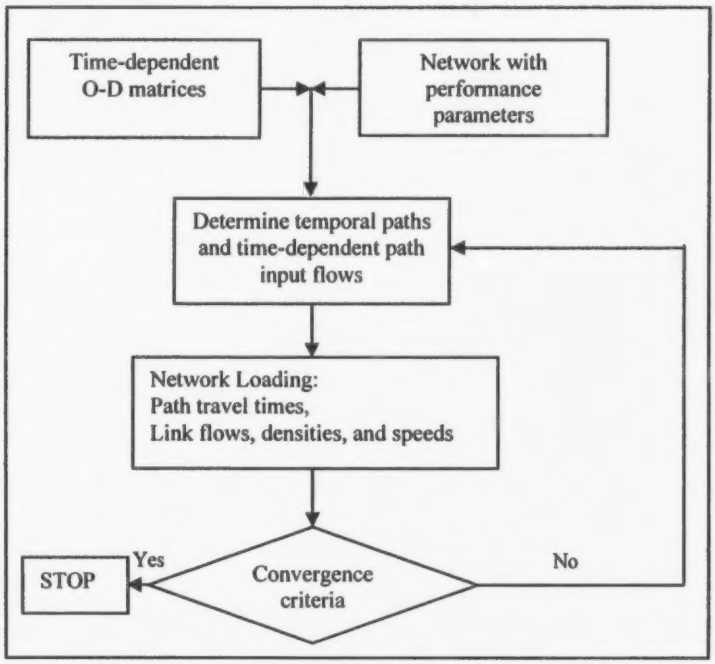

Figure 1. Structure of the solution algorithm for the DTA model

A Normalized (or Relative) Gap for each departure time interval τ may be computed as

$$RGap^{\tau}(n) = \frac{\sum_{i \in I} \sum_{k \in k_i} h_k^{\tau}(n) s_k^{\tau}(n) - \sum_{i \in I} g_i^{\tau} u_i^{\tau}(n)}{\sum_{i \in I} g_i^{\tau} u_i^{\tau}(n)} \tag{8}$$

where $u_i^{\tau}(n)$ are the lengths of the shortest paths at iteration n. A normalized gap of zero would indicate a perfect dynamic user equilibrium flow. Clearly this is a fleeting goal to aim for with any dynamic traffic assignment.

It is very important to note that this model, even though its general formulation is very similar to flow based models, is in fact a discrete vehicle model. The network loading procedure, as realized by the event based simulation, moves **individual cars** on the links of the network. This is addressed in the next section.

5. THE NETWORK LOADING METHOD

In the simple case of a homogeneous traffic stream (all vehicles with the same physical characteristics) on one-lane link, the simplified car following model underlying the traffic simulator can be solved as follows:

$$T(n,0)=\max\left[T(n,0)+\frac{L}{V},\ T(n-1,0)+\left(\tau+\frac{\lambda}{V}\right),\ T\left(n-\frac{L}{\lambda},L\right)+\tau\left(\frac{L}{\lambda}\right)\right]$$

Where λ is the effective vehicle length, τ is the driver/vehicle response time, and n is the vehicle number by order of arrival to the lane. L is the length of the link (multiple of λ), V is the free-flow speed, $T(n,0)$ is the time at which vehicle n enters the link, and $T(n,L)$ is the time at which vehicle n exits the link. The above relationship demonstrates how the model is solved in continuous-time using an event-based algorithm. This model yields a steady state triangular flow/density relationship where $Q=1/(\lambda/V+\tau)$, $K=1/\lambda$ and $W=\lambda/\tau$, is the negative wave speed.

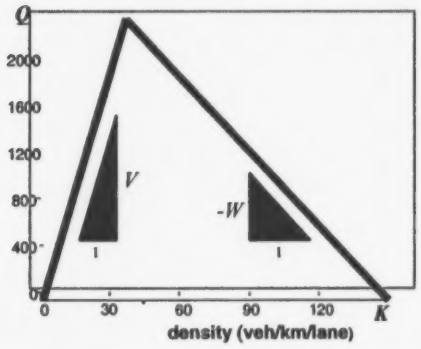

Figure 2. Flow-Density Relationship

The simulation, or car mover, that serves for the network loading, is a discrete-event procedure which moves **individual** vehicles through a network defined at the level of individual lanes. The computational effort per link required by this model is strictly proportional to the **number** of vehicles to pass through it, **regardless** of their travel times. Another special property of this model is that traffic dynamics are modeled **without** the (longitudinal) discretization of links into segments or cells. As a result, the procedure only explicitly calculates the time at which each vehicle crosses each node on its path. This leads to a **drastic** reduction in computational effort when compared to microscopic discrete time approaches, where the computational effort is a function of the total travel time experienced by the drivers. This is made possible by the results obtained by Mahut (2000) which demonstrate that the traffic dynamics obtained by the solution of the car following model used do not require the (longitudinal) discretization of links.

The underlying mechanism of congestion in the model is the crossing, merging and diverging of vehicle trajectories. Simply stated, whenever two vehicles pass the same point in space there must be a minimum time separation between them. Conflicts may occur on links, such as weaving movements, or at nodes, due to the interference of vehicles among

unprotected movements. Vehicle conflicts are resolved in traffic simulation models by gap-acceptance rules (Barcelo et al., 1994) (Van Aerde, 1999), which are based on one of the two vehicles having priority over the other, and the specification of a time-gap parameter. In continuum traffic models, the approach is to specify the maximum low-priority flow as a function of the prevailing high-priority flow (Leonard et al., 1999). In the model used here, a relatively simple gap-acceptance model has been implemented to determine precedence between vehicle conflicts at nodes, while a FIFO (first-in-first-out) rule is applied on links.

Once conflicts are identified and resolved, the resulting delays are propagated upstream by the car-following model. The *amount* of delay propagated from one vehicle to the next is exactly as would be determined by a standard queuing approach. What is different is *where* and *when* a vehicle in queue experiences each of the delays (or residuals thereof) that are propagated from downstream. This difference is due to the fact that the model employed here rigorously respects the finite speed at which delays propagate in actual traffic, sometimes called the negative wave speed. The positive (forward) wave speed is given by the speed limit. Mahut (2000) provides a detailed description of the model.

The pre-trip path information is complemented by a set of lane choice rules in order to provide the necessary information to identify conflicts between vehicle trajectories. In this way one can capture queue *spil- back* upstream and queue *spill-over* to adjacent lanes. The addition of look-ahead rules based strictly on local information has been shown to significantly improve the reality of the model outputs for some specific though not uncommon network topologies (Barceló, 2000) (Ben Akiva et al., 2000).

A problem associated with traffic control is that of preventing *gridlock*, or *deadlock*, in traffic networks. This situation occurs when a sequence of stopped vehicles forms a cycle in a network, and thus each driver is ultimately waiting for his/her own vehicle to move. This problem was addressed by developing an adaptive deadlock prevention algorithm that identifies when there is a high risk of a deadlocked cycle occurring. The algorithm explores the network starting at any given node using a depth-first search and continues as long as certain conditions are met. The additional computational effort is minimal.

The vehicle time update mechanism is done as follows:

- Each update of a vehicle at (upstream of) a node consists of the following:
  - next link determined by the path
  - arrival and departure lanes on next link:
    - look-ahead rules followed by local rules
  - calculation of time at which vehicle may execute movement

- After all vehicles have been updated at a node, select the next vehicle to move at the node (next event):
  - dead-lock avoidance algorithm:
    - gives priority to vehicles on high density cycles
  - determine precedence among conflicting vehicles
    - gap-acceptance modeling

In the next section some applications of this DTA model are reported.

6. APPLICATIONS

This dynamic traffic assignment algorithm was coded in C++ using an object-oriented approach. The code runs under Linux and Win XP/2000. The model was first applied in Stockholm and then in Calgary, Canada. The Calgary application and model calibration was reported in Mahut et al (2004). Other applications have been accomplished in Basel, Switzerland, in California, U.S.A., in Montreal and elsewhere. We give below the relative sizes of some of these applications and the computing times per iteration on an 1.6 Ghz Intel Centrino notebook computer operating under Win XP.

City	Zones	Nodes	Links	No. of vehicles	Time/iteration
Stockholm	114	1,191	2,080	108,000	1.5 min
Calgary	76	402	752	12,563	6.4 sec
Basel	62	376	642	47,768	10.7 sec
Montreal	232	966	2,563	232,667	7.5 sec
Bakersfield	64	368	582	47,023	51.2 sec

We present next some details of the Montreal application. The static planning model for the Montreal Region consists of 1,400 zones, 15,000 nodes and 33,500 links. A sub-area identified in pink in Figure 3 indicates the corridor that was identified for the study of an improved urban arterial facility. The demand was obtained by computing a traversal matrix for the sub-area by using the static model and then adjusting it by using link counts. More detail was added where necessary and some 320 traffic signals were coded by the staff of the City of Montreal. The resulting model has 232 zones, 966 nodes, 2,563 links and the total vehicle demand is 232,667. Three classes of vehicles were used: private cars, light trucks and heavy trucks. The data collection, coding and calibration effort took some nine months. The results of the calibration are shown in Figure 4 and a typical convergence result for 10 time intervals for the 6:30 – 9:30 AM peak period are shown in Figure 5. The calibration results are nearly as good as those obtained for the regional static model. Figure 6 shows a snapshot of a dynamic assignment results animation and Figure 7 is a snapshot of a scenario comparison. The facility that is improved is indicated with an arrow.

Figure 3. The Notre Dame sub-area

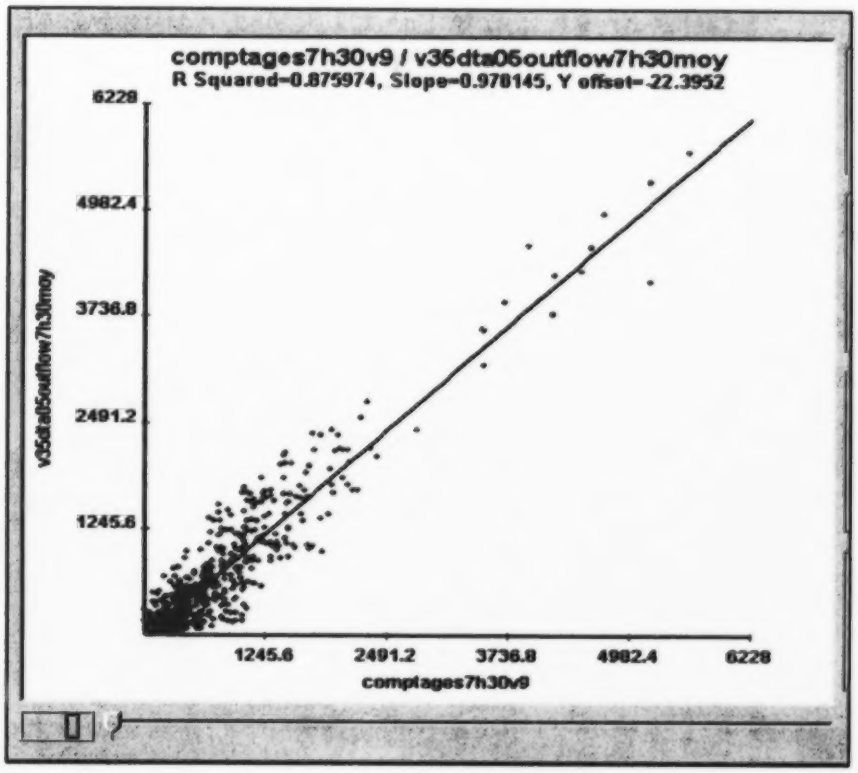

Figure 4. Scattergram for the AM Period

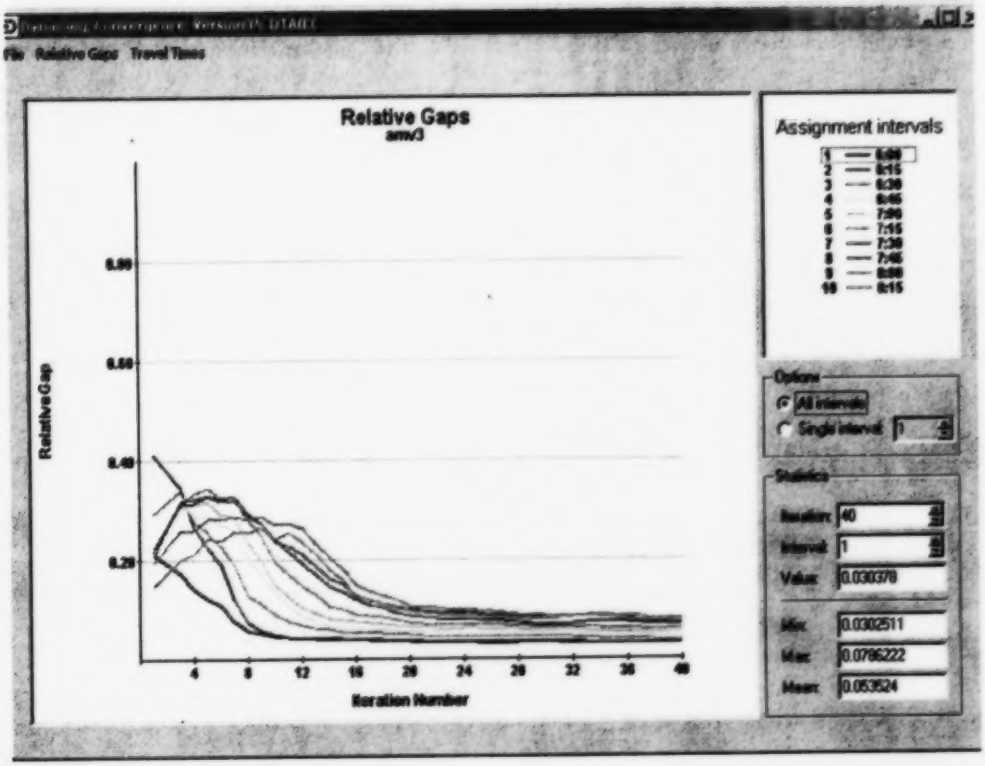

Figure 5. AM period convergence

Figure 6. AM flows colored by occupancy

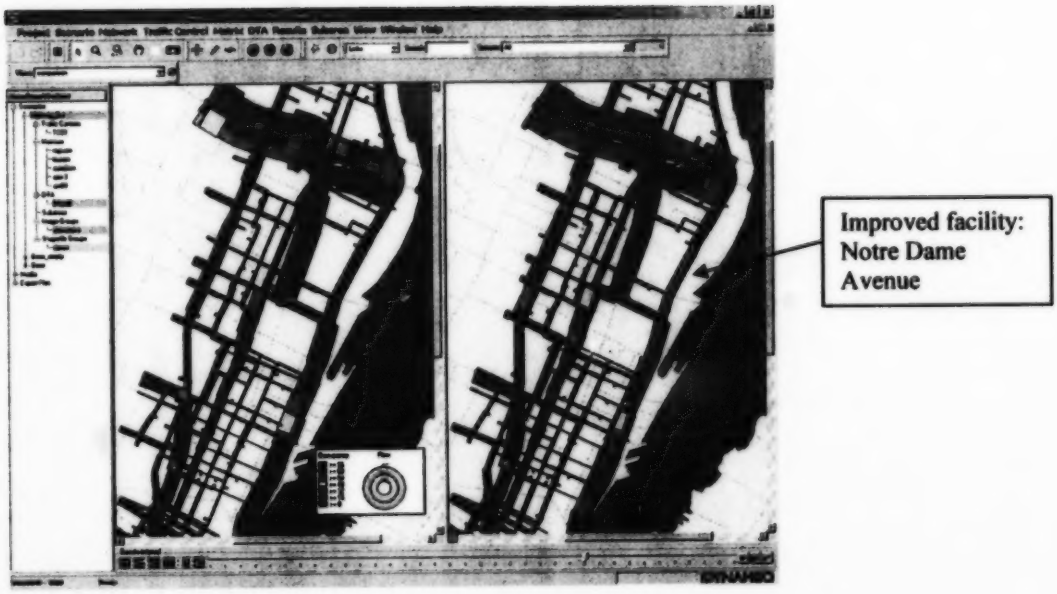

Figure 7. Scenario comparison : left window after improvement

7. CONCLUSIONS

A dynamic traffic assignment model, which uses a method of successive averages to determine pre-trip dynamic equilibrium path choices combined with an event-based traffic simulation model, was successfully applied to several medium-sized networks. The results indicate that an acceptable level of convergence can now be obtained for a medium-size network, using a realistic traffic model with a reasonable amount of computing time and memory usage.

The method has excellent potential for use in practice for a variety of applications related to the testing of ITS measures off-line. The model may have potential for further development as an on-line tool, due to the low computation times and memory requirements. Its computational efficiency is at least one order of magnitude faster than microscopic traffic simulation models. One of the next developments is the adaptation of the method for *en route* assignment and route guidance/

8. REFERENCES

Astarita, V., Er-Rafia, K., Florian, M., Mahut, M., Velan, S. (2001). Comparison of Three Methods for Dynamic Network Loading. *Transportation Research Record*, 1771:179-190.

Barceló, J. (2000). The Role of Traffic Simulation in Advanced Traffic Management Systems. Presented at the Spring meeting of INFORMS, Salt Lake City, USA. May 7-10, 2000.

Barceló, J., Ferrer, J.L., and Grau, R. (1994). AIMSUN2 and the GETRAM Simulation Environment. *Internal Report*, Departamento de Estadistica e Investigacion Operativa. Universitat Politecnica de Catalunya. See also http://www.tss-bcn.com.

Ben-Akiva, M., Koutsopoulis, H., Toledo, T. (2000). MITSIMLab: Recent Developments & Applications. Presented at the Spring meeting of INFORMS, Salt Lake City, USA. May 7-10, 2000.

Ben-Akiva, M., Koutsopoulos, H.N., Mishalani, R. (1998). "DynaMIT: A Simulation-Based System for Traffic Prediction". Paper presented at the DACCORD Short Term Forecasting Workshop, Delft, The Netherlands. See also its.mit.edu.

Diakaki, C. and Papageorgiou, M. (1996). Integrated Modelling and Control of Corridor Traffic Networks usingthe METACOR Modelling Tool. Dynamic Systems and Simulation Laboratory, Technical University of Crete. Internal Report No. 1996-8. Chania, Greece. pp. 41.

Dow, P. and Van Vliet, D. (1979). Capacity Restrained Road Assignment. *Traffic Eng. Control*, 20: 296-305.

Friesz, T., Bernstein, D., Smith, T., Tobin, R., Wie, B. (1993). A variational inequality formulation of the dynamic network user equilibrium problem. *Operations Research*, 41:179-191.

Lawphongpanich, S. and Hearn, D.W. (1984). Simplicial Decomposition of the Asymmetric Traffic Assignment Problem. Transportation Research B, 17: 123-133

Lighthill, M.J. and Whitham, G.B. (1955). On kinematic waves I: Flood movement in long rivers. II: A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London*, A229:281-345.

Leonard, D.R., Gower, P. and Taylor, N.B. (1989). CONTRAM: Structure of the Model. *Transport and Road Research Laboratory (TRRL) Research Report 178*. Department of Transport, Crowthorne. See also http://www.contram.com/.

Mahmassani, H.S., Abdelghany, A.F., Huynh, N., Zhou, X., Chiu, Y.-C. and Abdelghany, K.F. (2001). DYNASMART-P (version 0.926) User's Guide. Technical Report STO67-85-PIII, Center for Transportation Research, University of Texas at Austin.

Mahut, M. (2000). Discrete flow model for dynamic network loading. Ph.D. Thesis, Département d'informatique et de recherhe opérationelle, Université de Montréal. Published by the Center for Research on Transportation (CRT), University of Montreal.

Mahut, M., Florian, M., Tremblay, N., Campbell, M., Patman, D. and McDaniel., Z. (2004) Calibration And Application Of A Simulation-Based Dynamic Traffic Assignment Model, *Transportation Research Record* 1876: 101-111.

Messmer, A. (2000a). METANET A Simulation Program for Motorway Networks (Documentation). Dynamic Systems and Simulation Laboratory, Technical University of Crete. Chania, Greece.

Messmer, A (2000b). METANET-DTA An Exact Dynamic Traffic Assignment Tool Based on METANET. Dynamic Systems and Simulation Laboratory, Technical University of Crete. Chania, Greece. pp. 37.

Nagel, K. and Schreckenberg, M. (1992). A cellular automaton model for freeway traffic. *Journal de Physique I France*, 2:2221-2229.

Papageourgiou, M. (1990). Dynamic Modelling, Assignment and Route Guidance in Traffic Networks. Transportation Research, 24B(6), 471-95.

Richards, P.I. (1956). Shock waves on the highway. *Operations Research*, 4:42-51.

Van Aerde, M. (1999). INTEGRATION Release 2.20 for Windows: User's Guide. MVA and Associates, Kingston, Canada.

Yang, Q. (1997). A Simulation Laboratory for Evaluation of Dynamic Traffic management Systems. Ph.D. Thesis, Massachusetts Institute of Technology. See also http://web.mit.edu/its/products.html.

Ziliaskopulos, A.K. and Lee, S. (1997). A Cell Transmission Based Assignment-Simulation Model for Integrated Freeway/Surface Street Systems. *Transportation Research Record* 1701, 12-23.